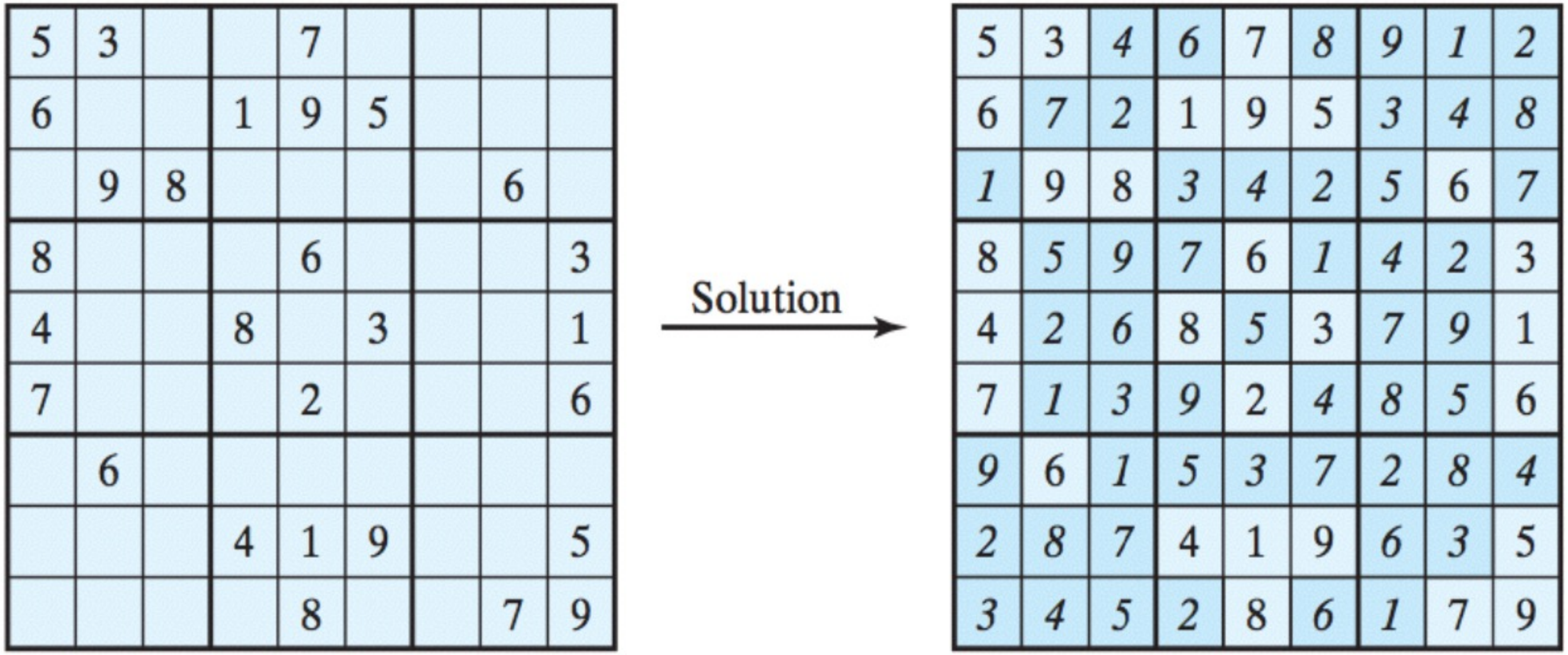
EXERCISE 1

Go as far as you can!

12.1



5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6							
			4	1	9			5
				8			7	9

(a) Puzzle

5	3	*4*	*6*	7	*8*	*9*	*1*	*2*
6	*7*	*2*	1	9	5	*3*	*4*	*8*
1	9	8	*3*	*4*	*2*	*5*	6	*7*
8	*5*	*9*	*7*	6	*1*	*4*	*2*	3
4	*2*	*6*	8	*5*	3	*7*	*9*	1
7	*1*	*3*	*9*	2	*4*	*8*	*5*	6
9	6	*1*	*5*	*3*	7	*2*	*8*	*4*
2	*8*	*7*	4	1	9	*6*	*3*	5
3	*4*	*5*	*2*	8	*6*	*1*	7	9

(b) Solution

Sudoku Problem

Sudoku is a 9 × 9 grid divided into smaller 3 × 3 boxes (also called regions or blocks), as shown in Figure (a). Some cells, called *fixed cells*, are populated with numbers from 1 to 9. The objective is to fill the empty cells, also called *free cells*, with numbers 1 to 9 so that every row, every column, and every 3 × 3 box contains the numbers 1 to 9, as shown in

12.2 (**IllegalTriangleException**) Define a **TriangleWithException** class with three sides. In a triangle, the sum of any two sides is greater than the other side. The **TriangleWithException** class must adhere to this rule. Create the **IllegalTriangleException** class, and modify the constructor of the **TriangleWithException** class to throw an **IllegalTriangleException** object if a triangle is created with sides that violate the rule, as follows:

```
/** Construct a triangle with the specified sides */
public TriangleWithException(double side1, double side2, double side3)
                    throws IllegalTriangleException {
    // Implement it
}
```

Implement a **TestTriangleWithException** class to test **TriangleWithException**.

12.3 (**NumberFormatException**) We have implemented the **hexToDecimal**(**String hexString**) method, which converts a hex string into a decimal number. Implement the **hexToDecimal** method to throw a **NumberFormatException** if the string is not a hex string. Test the **hexToDecimal** method in the main function.